



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/976,931	10/11/2001	Clifford L. Hersh	PA1951US	2047
<div>22830 7590 05/09/2007</div> <div>CARR & FERRELL LLP</div> <div>2200 GENG ROAD</div> <div>PALO ALTO, CA 94303</div>				
			<div>EXAMINER</div> <div>BULLOCK JR, LEWIS ALEXANDER</div>	
			<div>ART UNIT</div> <div>2195</div>	<div>PAPER NUMBER</div>
			<div>MAIL DATE</div> <div>05/09/2007</div>	<div>DELIVERY MODE</div> <div>PAPER</div>

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

09/976,931

Applicant(s)

HERSH, CLIFFORD L.

Examiner

Lewis A. Bullock, Jr.

Art Unit

2195

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 19 April 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-7, 10, 13, 15 and 17-28 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-7, 10, 13, 15 and 17-28 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Applicant's arguments, see after final response, filed April 19, 2007, with respect to the rejection(s) of claim(s) 1-7, 10, 13, 15, 17-28 under 35 U.S.C. 103 and finally rejected have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of "Group Update and Relaxed Balance in Search Trees" by VIRTANEN.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

3. Claims 1-7, 10, 13, 15 and 17-28 are rejected under 35 U.S.C. 102(a) as being anticipated by "Group Update and Relaxed Balance in Search Trees" by VIRTANEN.

As to claim 1, VIRTANEN teaches a method of reducing the number of times a tree data structure is rebalanced comprising the steps of: allowing a sub-tree of the tree data structure to grow until a number of unbalanced levels reaches a threshold greater than one (the absolute value of the height of the left minus the height of the right is less than or equal to one defines a balanced tree, otherwise the tree is unbalanced the one of the balancing acts is performed) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that

Art Unit: 2195

the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser.."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."); and rebalancing the tree data structure when the threshold (when the height of the left minus the height of the right is greater than one) is reached (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$.."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser.."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times.").

As to claim 2, VIRTANEN teaches a tree is balance if the height is equal to the log of the number of nodes (pg. 10-11, definition 2.6). It is inherent to the teachings of VIRTANEN that since the rebalance operation is performed when the height is not fulfilled (see page 14), that the tree is rebalanced when the height exceeds the log value.

As to claim 3, VIRTANEN teaches the threshold is a constant number of levels (greater than one via performing a rebalancing operation when the tree is not balanced

Art Unit: 2195

based absolute value of the height of the left minus the height of the right is less than or equal to one defines a balanced tree) greater than a level of a balanced portion of the tree data structure (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times.").

As to claim 4, VIRTANEN teaches the step of rebalancing the tree data structure comprises: developing first and second sets of rebalancing operation tasks (via uncoupled update and balance phases to delay / relax balancing) (see pg. 21-25); the first set of operation tasks operable to effect a first set of element transitions (via the locate and update phases of the index operation are tightly coupled and is to provide the updated information in the index as quickly as possible for future transactions to use) (pg. 22, Effects of Uncoupling Update and Balance Phases) and the second set of operation tasks (via the balance phases) operable to effect a second set of element state transitions (via balancing the tree by shifting / rotating nodes) (pg. 22, Effects of Uncoupling Update and Balance Phases), the first and second sets of element state transition being distinct one from the other (via being in distinct phases); performing the first set of operation tasks in a first phase; and performing the second set of operation

tasks in a second phase (via the updating being performed and the balancing being relaxed to a later phase) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."). It is also inherent to the teachings of VIRTANEN that the updating and rebalancing perform state transition between nodes by inserting / deleting nodes and shifting nodes.

As to claim 5, VIRTANEN teaches a method of deferring the rebalancing of a tree data structure comprising the steps of: allowing a sub-tree of the tree data structure to grow to an unbalanced length greater than one (the absolute value of the height of the left minus the height of the right is less than or equal to one defines a balanced tree, otherwise the tree is unbalanced the one of the balancing acts is performed) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."); and

Art Unit: 2195

rebalancing the tree data structure when the unbalanced length of the sub-tree reaches a threshold level (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times.").

As to claim 6 and 7, refer to claim 2 and 3 for rejection.

As to claim 10, VIRTANEN teaches a method of performing a rebalancing operation upon a tree data structure comprising the steps of: allowing a sub-tree of the tree data structure to grow unbalanced to a threshold level greater than one (the absolute value of the height of the left minus the height of the right is less than or equal to one defines a balanced tree, otherwise the tree is unbalanced the one of the balancing acts is performed) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-

Art Unit: 2195

logarithmic locate-complexity at all times."); developing, in the case where the sub-tree reaches the threshold level, first and second sets of rebalancing operation tasks (via uncoupled update and balance phases to delay / relax balancing) (see pg. 21-25), the first and second set of rebalancing operation tasks operable to effect a first and second set of element state transitions respectively (via the locate and update phases of the index operation are tightly coupled and is to provide the updated information in the index as quickly as possible for future transactions to use) (via balancing the tree by shifting / rotating nodes in the delayed balancing phase) (pg. 22, Effects of Uncoupling Update and Balance Phases); performing the first set of operation tasks in a first phase; and performing the second set of operation tasks in a second phase (via the updating being performed and the balancing being relaxed to a later phase) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."). It is also inherent to the teachings of VIRTANEN that the updating and rebalancing perform state transition between nodes by inserting / deleting nodes and shifting nodes.

As to claim 13, reference is made to a system that corresponds to the method of claim 1 and is therefore met by the rejection of claim 1 above.

As to claim 15, VIRTANEN teaches a system comprising: means for storing a tree data structure (database stores search tree) (pg. 3); means for tracking the execution of operations upon the tree data structure (via determining the absolute value of the height of the left minus the height of the right is less than or equal to one to determine if the tree is a balanced tree, otherwise the tree is unbalanced the one of the balancing acts is performed) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser.."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."); and means for rebalancing the tree data structure when an unbalanced sub-tree of the tree data structure reaches a threshold level greater than one (when the height of the left minus the height of the right is greater than one) is reached (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser.."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer

Art Unit: 2195

'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."), the rebalancing including a first rebalancing phase in which rebalancing operations are executed in parallel (via the locate and update phases of the index operation are tightly coupled and is to provide the updated information in the index as quickly as possible for future transactions to use) (pg. 22, Effects of Uncoupling Update and Balance Phases) and nodes of the unbalanced sub-tree are unlocked (see also pg. 30, 2nd paragraph, "...details of concurrency control are not included in the following discussion. Should two processes (either normal index operations or rebalancers) operate on the index structure simultaneously, they must involve a disjoint set of nodes..."; pg. 22, 3rd paragraph, "Hence, the rebalancing operation is well localized and can easily be performed while concurrently query processes work in other parts of the tree."), and a second rebalancing phase in which different rebalancing operations are executed (rebalancing operations) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."). It is also inherent to the teachings of VIRTANEN that the updating and rebalancing perform state transition between nodes by inserting / deleting nodes and shifting nodes.

As to claim 17, VIRTANEN teaches a method of deferring the rebalancing of a tree data structure comprising the steps of: tracking the performance of operations upon the tree data structure (via determining the absolute value of the height of the left minus the height of the right is less than or equal to one to determine if the tree is a balanced tree, otherwise the tree is unbalanced the one of the balancing acts is performed) (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."); and rebalancing the tree data structure when an unbalanced sub-tree of the tree data structure reaches a threshold level greater than one (when the height of the left minus the height of the right is greater than one) is reached (pg. 11, "An EBST is in balance if for all nodes t in T applies $|\text{height}(\text{left}(t)) - \text{height}(\text{right}(t))| \leq 1$."; pg. 22, "The evident approach is that the first two phases of an index operation, locate and update, are tightly coupled, but the balance phase will be delayed until the service requests become sparser."; pg. 25, "The database manager keeps track of the largest imbalance and triggers the rebalancer 'manually' as a predefined limit is exceeded, thus preserving a near-logarithmic locate-complexity at all times."), the rebalancing further comprising creating a first set of rebalancing operation tasks (via uncoupled update and balance phases to delay / relax balancing) (see pg. 21-25), the first set of rebalancing

Art Unit: 2195

operation tasks being characterized by navigation of the tree data structure using at least an existing link (via the locate and update phases of the index operation are tightly coupled and is to provide the updated information in the index as quickly as possible for future transactions to use by locating a node / position) (pg. 22, Effects of Uncoupling Update and Balance Phases), creating a second set of rebalancing operation tasks (rebalancing operations), the second set of rebalancing operation tasks being different from the first set of rebalancing operation tasks (via the indexing operations (e.g. the locate and update operations) are executed in one phase, the rebalancing operations are delayed / relaxed performed later) and being characterized by location of elements within the tree data structure using at least one pointer external to the tree data structure (via the use of group updating which uses an array / queue to retrieve stored update operations to be performed in a group) (see page 30-32 and pg. 34, 5.4 Balance Phase, "By processing the queued items in this exact order, the tree will be balanced in the proper manner...; pg. 35, procedure balanceRelaxed which has an input of the queue to balance) and created by the first set of rebalancing operation tasks (see, pg. 31, via the operation locate which stores items in the balancing queue / pg. 33, via the update phase adding information into the balancing queue to be processed), and performing at least one operation tasks of the first set of rebalancing operation tasks in a first phase and at least one second set of rebalancing operation tasks in a second phase (via the indexing operations (e.g. the locate and update operations) are executed in one phase, the rebalancing operations are delayed / relaxed performed later).

As to claim 18, refer to claim 17 for rejection.

As to claim 19, refer to claim 17 for rejection.

As to claim 20, VIRTANEN teaches executing the second set of rebalancing operation task is performed without navigating between nodes of the sub-tree (via performing the rebalancing on the nodes indicated in the queue and the use of group updating which uses the array / queue to retrieve stored update operations to be performed in a group) (see page 30-32 and pg. 34, 5.4 Balance Phase, "By processing the queued items in this exact order, the tree will be balanced in the proper manner...; pg. 35, procedure balanceRelaxed which has an input of the queue to balance) and created by the first set of rebalancing operation tasks (see, pg. 31, via the operation locate which stores items in the balancing queue / pg. 33, via the update phase adding information into the balancing queue to be processed).

As to claim 21, VIRTANEN teaches the execution of the first set of rebalancing operations includes generating a list of pointers to nodes require updating in the second phase of rebalancing (via performing the inserting operations and storing the information in the queue, thereby performing the rebalancing on the nodes indicated in the queue and the use of group updating which uses the array / queue to retrieve stored update operations to be performed in a group) (see page 30-32 and pg. 34, 5.4 Balance Phase, "By processing the queued items in this exact order, the tree will be balanced in

Art Unit: 2195

the proper manner...; pg. 35, procedure balanceRelaxed which has an input of the queue to balance) and created by the first set of rebalancing operation tasks (see, pg. 31, via the operation locate which stores items in the balancing queue / pg. 33, via the update phase adding information into the balancing queue to be processed)..

As to claim 22, refer to claim 17 for rejection.

As to claim 23, VIRTANEN teaches the first set of rebalancing operation tasks are performed on unlocked nodes of the sub-tree (see also pg. 30, 2nd paragraph, "...details of concurrency control are not included in the following discussion. Should two processes (either normal index operations or rebalancers) operate on the index structure simultaneously, they must involve a disjoint set of nodes..."; pg. 22, 3rd paragraph, "Hence, the rebalancing operation is well localized and can easily be performed while concurrently query processes work in other parts of the tree.").

As to claim 24, VIRTANEN teaches the first set of rebalancing operation tasks includes a plurality of operation tasks configured for parallel execution (see also pg. 30, 2nd paragraph, "...details of concurrency control are not included in the following discussion. Should two processes (either normal index operations or rebalancers) operate on the index structure simultaneously, they must involve a disjoint set of nodes..."; pg. 22, 3rd paragraph, "Hence, the rebalancing operation is well localized and

Art Unit: 2195

can easily be performed while concurrently query processes work in other parts of the tree.”).

As to claim 25, refer to claim 17 for rejection.

As to claim 26, VIRTANEN teaches the first set of rebalancing operation tasks are performed in parallel (see also pg. 30, 2nd paragraph, “...details of concurrency control are not included in the following discussion. Should two processes (either normal index operations or rebalancers) operate on the index structure simultaneously, they must involve a disjoint set of nodes...”; pg. 22, 3rd paragraph, “Hence, the rebalancing operation is well localized and can easily be performed while concurrently query processes work in other parts of the tree.”).

As to claim 27, refer to claim 26 for rejection.

As to claim 28, VIRTANEN teaches the first and second set of element state transitions each including changing pointers to nodes of the tree data structure (via updating phase which is part of the first set of element state transitions that changes the tree by inserting the node and the second set of element state transitions that rebalances the tree by shifting the nodes) (see page 32-32).

Claim Rejections - 35 USC § 112

4. Claims 2 and 6 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The cited claims detail about n nodes. The claims are not definitive in the amount of nodes of the tree data structure. The claim should read "n nodes" and not "about n nodes".

Conclusion

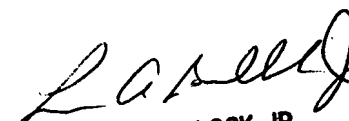
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (571) 272-3759. The examiner can normally be reached on Monday-Friday, 8:30 a.m. - 5:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2195

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

May 7, 2007


LEWIS A. BULLOCK, JR.
PRIMARY EXAMINER